



# Semi-nonnegative joint diagonalization by congruence and semi-nonnegative ICA

Julie Coloigner, Laurent Albera, Amar Kachenoura, Fanny Noury, Lotfi  
Senhadji

## ► To cite this version:

Julie Coloigner, Laurent Albera, Amar Kachenoura, Fanny Noury, Lotfi Senhadji. Semi-nonnegative joint diagonalization by congruence and semi-nonnegative ICA. *Signal Processing*, 2014, 105, pp.185-197. 10.1016/j.sigpro.2014.05.017 . inserm-01006599

**HAL Id: inserm-01006599**

**<https://www.hal.inserm.fr/inserm-01006599>**

Submitted on 26 Jun 2014

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Semi-nonnegative joint diagonalization by congruence and semi-nonnegative ICA

Julie Coloigner<sup>a,b</sup>, Laurent Albera<sup>a,b,c</sup>, Amar Kachenoura<sup>a,b</sup>, Fanny Noury<sup>a,b</sup>,  
Lotfi Senhadji<sup>a,b</sup>

<sup>a</sup>*Inserm, UMR 1099, Rennes, F-35000, France*

<sup>b</sup>*LTSI, Université de Rennes 1, Rennes, F-35000, France*

<sup>c</sup>*INRIA Rennes - Bretagne atlantique, Campus de Beaulieu, Rennes, F-35000, France*

---

## Abstract

In this paper, we focus on the Joint Diagonalization by Congruence (JDC) decomposition of a set of matrices, while imposing nonnegative constraints on the joint diagonalizer. The latter will be referred to the semi-nonnegative JDC fitting problem. This problem appears in semi-nonnegative Independent Component Analysis (ICA), say ICA involving nonnegative static mixtures, such as those encountered for instance in image processing and in magnetic resonance spectroscopy. In order to achieve the semi-nonnegative JDC decomposition, we propose two novel algorithms called ELS-ALS<sup>exp</sup> and CG<sup>exp</sup>, which optimize an unconstrained problem obtained by means of an exponential change of variable. The proposed methods are based on the line search strategy for which an analytic global plane search procedure has been considered. All derivatives have been jointly calculated in matrix form using the algebraic basis for matrix calculus and product operator properties. Our algorithms have been tested on synthetic arrays and the semi-nonnegative ICA problem is illustrated through simulations in magnetic resonance spectroscopy and in image processing. The numerical results show the benefit of using *a priori* information, such as non-negativity.

**Keywords:** Semi-nonnegative joint diagonalization by congruence,

---

*Email addresses:* julie.coloigner@gmail.com (Julie Coloigner),  
laurent.albera@univ-rennes1.fr (Laurent Albera)

## 1. Introduction and Problem formulation

The CANDECOMP/PARAFAC (CP) model, introduced by Harshman in the seventies, consists in expressing a  $q$ -th ( $q \geq 3$ ) order array as a minimal linear combination of  $q$ -th order rank-1 arrays [1]. This decomposition is very useful in many applications, such as in phonetics [1], in biomedical engineering [2] and in chemometrics [3]. More particularly, for a third-order array,  $\mathbf{X} \in \mathbb{R}^{I \times J \times K}$ , this model can also be expressed as a congruent diagonalization of a set of its  $K$  frontal slices, called  $\{\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(K)}\}$  as follows:

$$\forall k \in \{1, 2, \dots, K\}, \mathbf{X}^{(k)} = \mathbf{A}\mathbf{D}^{(k)}\mathbf{B}^\top + \mathbf{R}^{(k)} \quad (1)$$

with  $\mathbf{A} \in \mathbb{R}^{I \times P}$ ,  $\mathbf{B} \in \mathbb{R}^{J \times P}$  and  $\mathbf{R}^{(k)} \in \mathbb{R}^{I \times J}$  representing the model residual; and where the  $K$  matrices  $\mathbf{D}^{(k)} \in \mathbb{R}^{P \times P}$  are diagonal matrices.

In order to fit a set of symmetric matrices  $\{\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(K)}\}$ , Carroll and Chang [4] proposed INDividuals Differences in SCALing (INDSCAL) model as a special case of CP model (1) for three-way arrays that are symmetric in two modes i.e.  $\mathbf{A} = \mathbf{B}$ . It is also known as Joint Diagonalization by Congruence (JDC). It has been most often applied in psychometric literature [5], and more generally in data analysis, such as in the context of multiple factor analysis [6]. JDC model appears also in multivariate statistical signal processing and, more particularly, in Blind Source Separation (BSS) [7], whose problematic is to extract source components of interest from the observed signal. In this context, the matrix set  $\{\mathbf{D}^{(1)}, \dots, \mathbf{D}^{(K)}\}$  contains some statistical matrices of the sources, such as their correlation matrices at different lags, covariance matrices within different time intervals [8] or their different slice matrices of Higher Order statistics (HOS) [9].  $\{\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(K)}\}$  denotes the estimates of similar statistical matrices of the observed signals. Moreover, the statistical independence of the sources can be assumed in many areas, such as biomedical engineering

[10, 11], speech and audio [12] and spectroscopy [13] to cite a few, leading to the Independent Component Analysis (ICA) concept [7, 4]. Under this assumption, the statistical matrices  $\mathbf{D}^{(k)}$  are diagonal and the matrix  $\mathbf{A}$  represents the mixing matrix. Solving the JDC problem is more difficult than the CP ones (1) and most of the proposed algorithms imposed some restrictive conditions on the given matrices that can be diagonalized, or on the joint diagonalizer sought, such as orthogonality [8], positive-definiteness [14] or square joint diagonalizer [15]. If none of them is verified, JDC model is generally computed using a CP algorithm [16]. The two similar matrices are considered as distinct factors, without an explicit constraint enforcing equality. In practice, after convergence the two loading matrices will be equal. Ten Berge and Kiers show that in such cases this conjecture can be not valid [17].

Furthermore, some ICA applications involve a nonnegative mixing matrix, *i.e.* mixing matrix with nonnegative components, encountered, for instance, in image processing [18] or in Magnetic Resonance Spectroscopy (MRS) [19]. Indeed, each MRS observation is a linear combination of tissue or metabolite spectra, which are as independent as possible. The mixing matrix contains the positive weights representing concentrations. At present, in this context, the classical ICA algorithms, without nonnegative constraint, are used, in order to detect or classify tumors [20, 21]. Therefore, our primary motivation behind this paper is to consider such a constraint during the ICA process in order to improve the extraction quality as we will show in numerical simulations on simulated MRS data (see section 4). It gives rise to what we call semi-nonnegative ICA. This analysis can also be applied in image processing, to extract linear feature combinations, such as in facial recognition [22] or in prediction of rectal toxicity after radiotherapy [23]. Imposing the nonnegativity of the features allows us to facilitate interpretation and to obtain results in agreement with physical reality. To compute the semi-nonnegative ICA, the optimization problem we tackle is to fit the JDC model, while imposing nonnegative constraints on the joint diagonalizer,  $\mathbf{A}$ , as explained in section 3.5. The latter will be referred to the semi-nonnegative JDC fitting problem.

In this paper, the proposed methods fit explicitly the semi-nonnegative JDC model, taking into account symmetry and nonnegativity. The semi-nonnegative JDC model can be reformulated as a unique equation in a matrix form:  $\mathbf{X}_3 = \mathbf{C}(\mathbf{A} \odot \mathbf{A})^\top$  where the  $k$ -th row of  $\mathbf{C} \in \mathbb{R}^{K \times P}$  corresponds to the diagonal of  $\mathbf{D}^{(k)}$  and the symbol  $\odot$  stands for the Khatri-Rao product (column-wise Kronecker product). The matrix  $\mathbf{X}_3$  corresponds to the mode-3 unfolding of the corresponding three-way array  $\mathcal{X}$  [24]. To date, only two methods, briefly presented in [24, 25], allow to solve this problem. In [24, 26], as a way of including the nonnegative constraint in the JDC model, a square change of variable is used, such as each component  $(i, j)$  of  $\mathbf{A}$  is obtained by squaring the component  $(i, j)$  of a matrix  $\mathbf{B}$ . In this paper, we consider an exponential change of variable, like in [25], such as each component of the diagonalizer,  $\mathbf{A}$ , is the exponential of the component of a matrix  $\mathbf{E} \in \mathbb{R}^{I \times P}$ , denoted in matrix form for the sake of convenience  $\mathbf{A} = \exp(\mathbf{E})$ . To solve the semi-nonnegative JDC model, we propose to minimize the following Frobenius norm of the difference between a  $(K \times I^2)$  matrix  $\mathbf{X}_3$  and  $\mathbf{C}(\mathbf{A} \odot \mathbf{A})^\top$ , as follows:

$$\Psi(\mathbf{E}, \mathbf{C}) = \frac{1}{2} \|\mathbf{X}_3 - \mathbf{C}(\exp(\mathbf{E}) \odot \exp(\mathbf{E}))^\top\|_F^2 \quad (2)$$

where the couple of matrices  $(\mathbf{E}, \mathbf{C})$  belongs to the open set  $\mathbb{R}^{I \times P} \times \mathbb{R}^{K \times P}$ . To minimize this cost function, we develop two optimization procedures, which belong to the line search strategy, in section 3. They are based for the first one on an Alternating Least Square (ALS) procedure, called ELS-ALS<sup>exp</sup>, and for the second one on a nonlinear Conjugate Gradient (CG) descent, named CG<sup>exp</sup>. To accelerate the convergence, the two approaches are combined with a line search, described in section 3.3. The latter improves the performance of our ELS-ALS-like method briefly presented in [25], based on an iterative procedure of line search. Moreover, we provide compact matrix expressions of the derivatives and the parameters used in the two optimization procedures, which allow for a direct implementation of our iterative algorithms in matrix programming environments, like MATLAB. Note that the exponential change of variable prevents to process a matrix  $\mathbf{A}$  with some exact zero components. To overcome this, an

alternative strategy is incorporated in the two proposed methods, as described in section 3. The numerical complexity is calculated analytically in section 3.4. In section 4, numerical results show the benefit of our methods compared with non-constrained CP methods [27, 28], classical JDC algorithms [7] and semi-nonnegative JDC techniques [24]. Moreover, the concept of semi-nonnegative ICA is introduced in section 3.5. The semi-nonnegative ICA algorithms proposed in this paper exploit some interesting properties enjoyed by Second Order (SO), Third Order (TO) and Fourth Order (FO) cumulants in the presence of independent random processes and are based on our semi-nonnegative JDC optimization procedures. The behavior of our semi-nonnegative methods is compared to that of classical ICA methods [8] and an NMF algorithm [29] on MRS signals and image data.

## 2. Notations

The symbol  $\otimes$  denotes the Kronecker product while  $\odot$  and  $\boxtimes$  stand for the Khatri-Rao product (column-wise Kronecker product) and the Hadamard product (element-wise product), respectively. The operator  $\text{Mat}$  rearranges a block matrix (or vector) into another; such as  $\text{Mat}^{(N \times P, M)}([\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_M]^\top) = [\mathbf{A}_1^\top, \mathbf{A}_2^\top, \dots, \mathbf{A}_M^\top]$ , where the  $M$  blocks,  $\mathbf{A}_i$  with  $1 \leq i \leq M$ , are all of size  $(N \times P)$ . The symbol  $\text{diag}$  is the MATLAB operator, given diagonal matrix and diagonal of a matrix depending on the size of the argument. When  $\mathbf{z}$  is a vector,  $\text{diag}(\mathbf{z})$  stands for the  $(I \times I)$  diagonal matrix built from the  $I$ -dimensional vector  $\mathbf{z}$ . When  $\mathbf{Z}$  is a matrix of size  $(I \times I)$ ,  $\text{diag}(\mathbf{Z})$  is a column vector containing the  $I$  diagonal component of  $\mathbf{Z}$ . Furthermore, the superscripts  $^\#$  and  $^\top$  stand for the Moore-Penrose pseudo-inverse and the transpose operators, respectively. The  $(N \times N)$  identity matrix is denoted by  $\mathbf{I}_N$ . The permutation matrix,  $\mathbf{U}_{PN}$  of size  $(PN \times PN)$ , is defined by [30]:  $\mathbf{U}_{PN} = \sum_{p=1}^P \sum_{n=1}^N \mathbf{E}_{pn}^{(P \times N)} \otimes \mathbf{E}_{np}^{(N \times P)}$ , where  $\mathbf{E}_{pn}^{(P \times N)}$  is a  $(P \times N)$  elementary matrix of zeros except the  $(p, n)$ -th component which is set to one. The trace of the square matrix  $\mathbf{Z}$  is denoted by  $\text{Tr}(\mathbf{Z})$ . The  $N$ -dimensional vector of ones and the  $(N \times P)$ -dimensional matrix

of ones are denoted by  $\mathbf{1}_N$  and  $\mathbf{1}_{N \times P}$ , respectively. We denote:  $\mathbf{A}^{\odot 2} = \mathbf{A} \odot \mathbf{A}$  and  $\mathbf{B}^{\square 2} = \mathbf{B} \square \mathbf{B}$ . Finally  $|\cdot|$ ,  $\|\cdot\|$  and  $\|\cdot\|_F$  stand for the absolute value, the Euclidean norm and the Frobenius norm, respectively.

### 3. Nonnegative JDC Methods

#### 3.1. The nonlinear Conjugate Gradient (CG) method

The nonlinear CG method is widely used in practice, particularly to solve large-scale nonlinear optimization problems. The main advantage of this algorithm lies in the combination of small memory allocation and low numerical complexity. Moreover, it is characterized by an interesting convergence property [31].

At iteration  $it$ , the update rule of  $\mathbf{E}$  and  $\mathbf{C}$  of the  $\text{CG}^{\text{exp}}$  algorithm is given by:

$$\begin{bmatrix} \text{vec}(\mathbf{E}^{it+1}) \\ \text{vec}(\mathbf{C}^{it+1}) \end{bmatrix} = \begin{bmatrix} \text{vec}(\mathbf{E}^{it}) \\ \text{vec}(\mathbf{C}^{it}) \end{bmatrix} - \begin{bmatrix} \mu_{\mathbf{E}}^{it} \mathbf{I}_{IP} & \mathbf{0} \\ \mathbf{0} & \mu_{\mathbf{C}}^{it} \mathbf{I}_{KP} \end{bmatrix} \begin{bmatrix} \text{vec}(\mathbf{G}_{\mathbf{E}}^{it}) \\ \text{vec}(\mathbf{G}_{\mathbf{C}}^{it}) \end{bmatrix} \quad (3)$$

where  $\text{vec}(\mathbf{G}_{\mathbf{E}}^{it})$  and  $\text{vec}(\mathbf{G}_{\mathbf{C}}^{it})$  are the two descent directions and where the two stepsizes  $\mu_{\mathbf{E}}^{it}$  and  $\mu_{\mathbf{C}}^{it}$  are computed by a line search procedure (see section 3.3). These parameters are calculated analytically at each iteration by minimizing the objective function along  $\text{vec}(\mathbf{G}_{\mathbf{E}}^{it})$  and  $\text{vec}(\mathbf{G}_{\mathbf{C}}^{it})$ . In practice, the line search permits to accelerate the convergence and to sometimes avoid potential local minima. Moreover, in the CG method,  $\text{vec}(\mathbf{G}_{\mathbf{E}}^{it})$  and  $\text{vec}(\mathbf{G}_{\mathbf{C}}^{it})$  are a linear combination of the gradients and the previous descent directions and are updated at iteration  $it$  according to the following rule:

$$\begin{bmatrix} \text{vec}(\mathbf{G}_{\mathbf{E}}^{it}) \\ \text{vec}(\mathbf{G}_{\mathbf{C}}^{it}) \end{bmatrix} = - \begin{bmatrix} D_{\mathbf{E}}\Psi(\mathbf{E}^{it}, \mathbf{C}^{it})^{\top} \\ D_{\mathbf{C}}\Psi(\mathbf{E}^{it}, \mathbf{C}^{it})^{\top} \end{bmatrix} + \beta^{it-1} \begin{bmatrix} \text{vec}(\mathbf{G}_{\mathbf{E}}^{it-1}) \\ \text{vec}(\mathbf{G}_{\mathbf{C}}^{it-1}) \end{bmatrix} \quad (4)$$

where the gradients  $D_{\mathbf{E}}\Psi(\mathbf{E}^{it}, \mathbf{C}^{it})$  and  $D_{\mathbf{C}}\Psi(\mathbf{E}^{it}, \mathbf{C}^{it})$  are given in Lemma 1 (see Appendix A) and where  $\beta^{it-1}$  is computed using the Polak-Ribière formula [31]:

$$\beta^{it-1} = \frac{(D\Psi(\mathbf{E}^{it-1}, \mathbf{C}^{it-1}) - D\Psi(\mathbf{E}^{it-2}, \mathbf{C}^{it-2}))D\Psi(\mathbf{E}^{it-1}, \mathbf{C}^{it-1})^{\top}}{D\Psi(\mathbf{E}^{it-2}, \mathbf{C}^{it-2})D\Psi(\mathbf{E}^{it-2}, \mathbf{C}^{it-2})^{\top}} \quad (5)$$

with  $D\Psi(\mathbf{E}^{it}, \mathbf{C}^{it}) = [D_{\mathbf{E}}\Psi(\mathbf{E}^{it}, \mathbf{C}^{it}), D_{\mathbf{C}}\Psi(\mathbf{E}^{it}, \mathbf{C}^{it})]$ . Note that, the direc-

---

**Algorithm 1** Pseudo-code for  $\text{CG}^{\text{exp}}$

---

Initialize both matrices  $\mathbf{E}^0$  and  $\mathbf{C}^0$ .

Set  $it = 1$ , the relative error  $\text{err} = 1$ , the maximal number  $itmax$  of allowed iterations, the stop criterion threshold  $\epsilon$  and the restarting number  $n$ .

Calculate  $D_{\mathbf{E}}\Psi(\mathbf{E}^0, \mathbf{C}^0)$ ,  $D_{\mathbf{C}}\Psi(\mathbf{E}^0, \mathbf{C}^0)$  (see Lemma 1),  $\mu_{\mathbf{E}}^0$  and  $\mu_{\mathbf{C}}^0$  (see section 3.3) and compute  $\mathbf{E}^1$  and  $\mathbf{C}^1$  with a steepest descent step.

**while**  $\text{err} > \epsilon$  or  $it < itmax$  **do**

    Compute  $D_{\mathbf{E}}\Psi(\mathbf{E}^{it}, \mathbf{C}^{it})$  and  $D_{\mathbf{C}}\Psi(\mathbf{E}^{it}, \mathbf{C}^{it})$  using Lemma 1.

    If  $it$  is proportional to  $n$ ,  $\text{vec}(\mathbf{G}_{\mathbf{E}}^{it}) = D_{\mathbf{E}}\Psi(\mathbf{E}^{it}, \mathbf{C}^{it})^\top$  and  $\text{vec}(\mathbf{G}_{\mathbf{C}}^{it}) = D_{\mathbf{C}}\Psi(\mathbf{E}^{it}, \mathbf{C}^{it})^\top$ , else compute  $\beta^{it-1}$ , defined in (5), and update the direction  $[\text{vec}(\mathbf{G}_{\mathbf{E}}^{it})^\top, \text{vec}(\mathbf{G}_{\mathbf{C}}^{it})^\top]^\top$  using (4).

    Compute the two stepsizes  $\mu_{\mathbf{E}}^{it}$  and  $\mu_{\mathbf{C}}^{it}$  according to section 3.3.

    Compute the matrices  $\mathbf{E}^{it+1}$  and  $\mathbf{C}^{it+1}$  using the update rule (3).

    If  $\Psi(\mathbf{E}^{it+1}, \mathbf{C}^{it+1}) > \Psi(\mathbf{E}^{it}, \mathbf{C}^{it})$ , compute the matrices  $\mathbf{E}^{it+1}$  and  $\mathbf{C}^{it+1}$  again with a steepest descent step ( $\beta^{it-1} = 0$ ).

    Compute  $\text{err} = |\Psi(\mathbf{E}^{it+1}, \mathbf{C}^{it+1}) - \Psi(\mathbf{E}^{it}, \mathbf{C}^{it})| / \Psi(\mathbf{E}^{it}, \mathbf{C}^{it})$ .

    Set  $it = it + 1$ .

**end while**

---

tions  $\text{vec}(\mathbf{G}_{\mathbf{E}}^{it})$  and  $\text{vec}(\mathbf{G}_{\mathbf{C}}^{it})$  contain the informations of all previous iterates. It is noteworthy that periodically erasing old information may be beneficial. To do so, the direction is refreshed by simply taking a gradient direction, at every  $n$  iterations. Moreover, the stopping of the algorithm in the case of an ascent direction is replaced by a simple gradient descent and then the CG update is used again at the next iteration. Although the exponential function is not defined at zero, at the end of convergence, as the exponential of an element of  $\mathbf{E}$  is lower than a threshold value, close to zero, the associated component of  $\mathbf{A}$  is arbitrary fixed to zero. It allows us to process  $\mathbf{A}$  mixtures with some exact zero components. This pseudo-code is described in Algorithm 1.



### 3.2. The ELS-ALS method

#### 3.2.1. Alternating Least Square (ALS)

The second method is based on an ALS procedure, in order to minimize (2) due to its advantages brought to light in the context of the CP decomposition. Bro and Tomasi [3] found that this optimization scheme has good results in terms of decomposition accuracy, numerical complexity and memory requirements. Its principle is to reduce the non-linear minimization problem of  $\Psi$  (2) to several coupled linear least square subproblems. To do so, the cost function (2) is alternatively minimized w.r.t.  $\mathbf{E}$  and  $\mathbf{C}$ . Then, we solve sequentially the two following subproblems:

$$\mathbf{E}^{it+1} = \arg \min_{\mathbf{E}} \Psi_E = \arg \min_{\mathbf{E}} \|\mathbf{X}_3 - \mathbf{C}^{it} (\exp(\mathbf{E}) \odot \exp(\mathbf{E}))^\top\|_F^2 \quad (6)$$

$$\mathbf{C}^{it+1} = \arg \min_{\mathbf{C}} \Psi_C = \arg \min_{\mathbf{C}} \|\mathbf{X}_3 - \mathbf{C}(\mathbf{A}^{it+1} \odot \mathbf{A}^{it+1})^\top\|_F^2 \quad (7)$$

with  $\mathbf{A}^{it+1} = \exp(\mathbf{E}^{it+1})$  and where  $\Psi_E$  and  $\Psi_C$  depend only on one free matrix variable,  $\mathbf{E}$  and  $\mathbf{C}$ , respectively. The two minimization problems can be solved by vanishing the matrix partial derivatives, given in Lemma 1 (see Appendix A). Concerning the subproblem (6) w.r.t.  $\mathbf{E}$ , due to the complexity of gradient, we are not able to find an analytical matrix zero. Then, we alternatively minimize  $\Psi_E$  w.r.t. each component of  $\mathbf{E}$ , as described in Appendix B. In addition the proposed procedure allows us to process mixtures with some exact zero components, while the exponential change of variable prevents that. The solution of the minimization of  $\Psi_C$  (7) is well-known and given by  $\mathbf{C} = \mathbf{X}_3((\mathbf{A} \odot \mathbf{A})^\#)^\top$  [28].

#### 3.2.2. Enhanced Line Search (ELS)

The ALS procedure has some known drawbacks such as a slow convergence in the context of ill-conditioned factors or high collinearity between factors, and a sensitivity to initialization. The ALS procedure can stay trapped in a local minimum for several iterations. The ELS procedure, which will be described in detail in section 3.3 seeks to calculate the optimal stepsizes to exit faster from local minima and thus to accelerate the ALS algorithm convergence.

The direction used during the ELS step is most of the times a linear direction equal to the difference between two successive iterates. However, Comon et al. [28] showed that the ELS cannot improve the convergence in case of bad linear direction. One possibility is to define a new search direction based on a extrapolation. Chen et al. [32] proposed such a search direction containing higher-order information, which is a linear combination of three previous iterates, depending on an empirical positive parameter  $\tau \in [0, 1]$  to set:

$$\mathbf{G}_M^{it} = (1 + 2\tau)\mathbf{M}^{it+1} - (1 + 3\tau)\mathbf{M}^{it} + \tau\mathbf{M}^{it-1} \quad (8)$$

with  $\mathbf{M} = \{\mathbf{E}, \mathbf{C}\}$  and where  $\mathbf{M}^{it+1}$ ,  $\mathbf{M}^{it}$  and  $\mathbf{M}^{it-1}$  are the matrix estimates in the ALS iteration  $it + 1$ ,  $it$  and  $it - 1$ . Then, at iteration  $it$ , after the ALS step, the update rule characterizing the ELS strategy, can be defined as follows:

$$\mathbf{E}^{new} = \mathbf{E}^{it-1} + \mu_E^{it} \mathbf{G}_E^{it} \quad \mathbf{C}^{new} = \mathbf{C}^{it-1} + \mu_C^{it} \mathbf{G}_C^{it} \quad (9)$$

Both stepsizes,  $\mu_E^{it}$  and  $\mu_C^{it}$ , can be computed by minimizing the line search cost function along the cycle directions, as described in section 3.3. Then, to have a good compromise between effectiveness and numerical complexity, it is better to calculate the optimal stepsizes, every  $k$  iterations with  $k > 1$ . Then,  $\mathbf{E}^{new}$  and  $\mathbf{C}^{new}$  are the matrices that will be used in the ALS step at iteration  $it + 1$  instead of  $\mathbf{E}^{it}$  and  $\mathbf{C}^{it}$ . The pseudo-code of ELS-ALS<sup>exp</sup> is given in algorithm 2.

### 3.3. The computation of line search

It consists of searching stepsizes,  $\mu_E^{it}$  and  $\mu_C^{it}$ , minimizing the cost function  $\Psi$  (2) along the directions,  $\mathbf{G}_E^{it}$  and  $\mathbf{G}_C^{it}$ , as follows:

$$\varphi(\mu_E, \mu_C) = \left\| \mathbf{X}_3 - (\mathbf{C}^l + \mu_C \mathbf{G}_C^{it}) ((\exp(\mathbf{E}^{it} + \mu_E \mathbf{G}_E^{it}))^{\odot 2})^\top \right\|_F^2 \quad (10)$$

w.r.t.  $\mu_E$  and  $\mu_C$ , with  $l = it$  for CG<sup>exp</sup> and  $l = it - 1$  for ELS-ALS<sup>exp</sup>. This procedure of stepsize calculation is common for the two semi-nonnegative JDC algorithms. It differs only on the choice of  $\mathbf{G}_E^{it}$  and  $\mathbf{G}_C^{it}$ , given by (4) (resp. (8)) for CG<sup>exp</sup> (resp. ELS-ALS<sup>exp</sup>).

---

**Algorithm 2** Pseudo-code for ELS-ALS<sup>exp</sup>

---

Initialize both matrices  $\mathbf{B}^0$  and  $\mathbf{C}^0$ .

Set  $it = 0$ ,  $k = 5$  and the relative error  $\text{err} = 1$  and set the maximal number  $itmax$  of allowed iterations and the stopping criterion threshold  $\epsilon$ .

**while**  $\text{err} > \epsilon$  or  $it < itmax$  **do**

    Compute each element of  $\mathbf{E}^{it+1}$ , as explained in section (3.2).

    Compute  $\mathbf{C}^{it+1}$ , as explained in section (3.2).

    if  $\text{mod}(it, k) = 0$ , update the directions  $\mathbf{G}_E^{it}$  and  $\mathbf{G}_C^{it}$  (8). Compute the stepsizes  $\mu_E^{it}$  and  $\mu_C^{it}$  using the global plane search scheme, described in section 3.3, and update the matrices  $\mathbf{E}^{it+1}$  and  $\mathbf{C}^{it+1}$  using (9).

    Set  $it = it + 1$ .

**end while**

---

The line search procedure can be done approximately or exactly [1, 33, 28, 26]. Regarding the semi-nonnegative JDC model, Coloigner et al. [24, 34] developed an exact line search procedure, for an ALS algorithm based on a square change of variable. However, since the exponential function is not homogeneous, the exact minimization of (10) leads to root a function which is the composite of a polynomial and the exponential function. So, the problem we tackle has no analytical root. In [25], we have calculated the zeros of the derivative function  $\varphi'$  by MATLAB rooting. Unfortunately, the procedure may be trapped in local extrema and so the global minimum may be not found. To overcome this difficulty, we propose to approximate the exponential by its truncated power serie. Indeed, the latter allows us to approximate  $\varphi$  by a polynomial in  $\mu_E$  and  $\mu_C$  and to calculate the global minimum analytically. By choosing a sufficiently large expansion order we can make the approximation as accurate as we wish. However, the choice of a large order increases considerably the degree of the polynomial. For instance, we have a 35-th degree with a third order truncated power series, a 47-th degree polynomial for a fourth order... A high degree leads to a high numerical complexity and problems of root stability. To obtain accurate results combining a relatively low numerical complexity and a

good stability, we propose to use a third order truncated power series of the exponential function and the criterion (10) can be reformulated as follows:

$$\varphi(\mu_E, \mu_C) \approx \left\| \mathbf{X}_3 - (\mathbf{C}^l + \mu_C \mathbf{G}_C^{it})(\exp(\mathbf{E}^l) \boxtimes \sum_{n=0}^3 \frac{(\mu_E \mathbf{G}_E^{it})^{\boxtimes n}}{n!})^{\odot 2} \right\|_F^2 \quad (11)$$

w.r.t.  $\mu_E$  and  $\mu_C$ . Note that equation (11) can be reduced to a compact form as follows:

$$\varphi(\mu_E, \mu_C) \approx \|\mathbf{F}\mathbf{u}\|_F^2 = \mathbf{u}^\top \mathbf{F}^\top \mathbf{F} \mathbf{u} = \mathbf{u}^\top \mathbf{Q} \mathbf{u} \quad (12)$$

where  $\mathbf{F}$  is a  $(I^2 K \times 14)$  matrix (see Appendix C), where  $\mathbf{u} = [\mu_C \mu_E^6, \mu_C \mu_E^5, \mu_C \mu_E^4, \mu_C \mu_E^3, \mu_C \mu_E^2, \mu_C \mu_E, \mu_C, \mu_E^6, \mu_E^5, \mu_E^4, \mu_E^3, \mu_E^2, \mu_E, 1]^\top$  is a 14-dimensional vector and where  $\mathbf{Q}$  is a symmetric matrix. To reduce the numerical complexity,  $\mathbf{Q}$  can be directly calculated, without computing  $\mathbf{F}$ , using the Khatri-Rao property [30] (see Appendix D). After developping (12), the objective function  $\varphi$  is a second degree polynomial in  $\mu_C$ . Thus, the optimal stepsize  $\mu_C^{it}$  is a rational function in  $\mu_E$ . Once  $\mu_C^{it}$  is computed, its expression is injected in the equation  $\partial\varphi/\partial\mu_E = 0$ , which is a 35-th degree polynomial in  $\mu_E$ . Then, the optimal stepsize  $\mu_E^{it}$  is computed as the root of this polynomial, allowing us afterward to calculate  $\mu_C^{it}$ .

### 3.4. Numerical complexity

This section is devoted to compute the numerical complexity, which is defined as the number of floating point operations (flop) required to execute a iteration. A flop corresponds to a multiplication followed by a addition. Note that all expressions have been calculated analytically. The two methods, CG<sup>exp</sup> and ELS-ALS<sup>exp</sup>, require the computation of stepsizes, which are obtained by a line search procedure, as described in section 3.3. It costs approximately  $O(I^2 KP + P^2(I+K))$  flops. Note that the first term  $I^2 KP$  is associated with the computation of the last row and column of the matrix  $\mathbf{Q}$  (12) and the second term  $P^2(I+K)$  corresponds to the other simplified components (see Appendix D).

For CG<sup>exp</sup>, besides the computation of stepsizes, each iteration involves the ones of the gradient. The cost of the gradient is the sum of both complexities of

$D_{\mathbf{E}}\Psi(\mathbf{E}, \mathbf{C})$  and  $D_{\mathbf{C}}\Psi(\mathbf{E}, \mathbf{C})$  (see Lemma 1 in Appendix A).  $D_{\mathbf{E}}\Psi(\mathbf{E}, \mathbf{C})$  and  $D_{\mathbf{C}}\Psi(\mathbf{E}, \mathbf{C})$  approximately amount to  $O(I^2P(P^3+IP+K))$  and  $O(KP(KP+I^2))$  flops, respectively. This algorithm costs approximately  $O(I^3P^2+I^2(P^4+KP)+K^2P^2)$ .

At each iteration, ELS-ALS<sup>exp</sup> algorithm requires the element by element update of matrix  $\mathbf{E}$ , which approximately amounts to  $O(I^3(KP^2+P^3)+IKP^3)$  multiplications. This step consists of calculating the coefficients  $\beta_1, \beta_2$  and  $\beta_4$  of the polynomial  $\Phi'_{E_{a,b}}$  (B.2) as well as the computation of the stationary points and the selection of the positive minimum, as described in section Appendix B. The estimate of the matrix  $\mathbf{C}$  is the well-known solution of the CP model and its cost is approximately  $O(KP(I^2+P))$  flops by assuming that  $P \leq I^2$  [28]. In addition, to accelerate the convergence, the line search step is used at every  $k$  iterations and its cost is given previously. This algorithm costs approximately  $O(I^3(KP^2+P^3)+IKP^3)$  flops.

### 3.5. Application to semi-nonnegative ICA

In this section, we introduce the semi-nonnegative ICA problem, defined as follow:

Given one realization of a real random vector process  $\{\mathbf{x}[m]\}$ , find an  $(I \times P)$  mixing matrix  $\mathbf{A}$  and one realization of a  $P$ -dimensional source random process  $\{\mathbf{s}[m]\}$  such that for each index  $m$ ,  $\mathbf{x}[m] = \mathbf{A}\mathbf{s}[m] + \mathbf{e}[m]$  where  $\mathbf{A}$  has nonnegative components,  $\mathbf{s}[m]$  has statistically independent components, and  $\mathbf{e}[m]$  is an  $I$ -dimensional Gaussian noise vector, independent of  $\mathbf{s}[m]$ .

This problem turns up in some ICA applications, which involve nonnegative static mixtures, such as in image processing [21] and in MRS applications [35].

Furthermore, the independence property lead to several discussions, in some ICA applications. In MRS, for instance, spectra of different chemical pure substances may show very large overlaps and have nonnegative components, which may involve partial dependence between some spectra. In practice, the classical ICA without nonnegative constraints may succeed in separating non totally independent sources, but the nonnegativity property is not ensured in this case.

The idea throughout this paper is then to enforce the nonnegativity property of the mixing matrix during the ICA procedure, giving results in agreement with physical reality in such context.

The semi-nonnegative ICA algorithms can exploit some interesting properties enjoyed by the Second Order (SO), Third Order (TO) and Fourth Order (FO) cumulants in the presence of mixed independent random processes and the optimization procedures described in section 3. We propose in this section the problem to show how we can combine SO, TO and FO cumulants into a set of symmetrical matrices following the semi-nonnegative JDC model. However, the user can choose also only one or two cumulants.

In BSS context, under the assumption of almost independent sources and the multi-linearity property of cumulants, we get:

$$\mathcal{C}_{n_1, n_2, \mathbf{x}} = \sum_{p=1}^P A_{n_1, p} A_{n_2, p} \mathcal{C}_{p, p, \mathbf{s}} + R_{n_1, n_2, \mathbf{x}} \quad (13)$$

$$\mathcal{C}_{n_1, n_2, n_3, \mathbf{x}} = \sum_{p=1}^P A_{n_1, p} A_{n_2, p} A_{n_3, p} \mathcal{C}_{p, p, p, \mathbf{s}} + R_{n_1, n_2, n_3, \mathbf{x}} \quad (14)$$

$$\mathcal{C}_{n_1, n_2, n_3, n_4, \mathbf{x}} = \sum_{p=1}^P A_{n_1, p} A_{n_2, p} A_{n_3, p} A_{n_4, p} \mathcal{C}_{p, p, p, p, \mathbf{s}} + R_{n_1, n_2, n_3, n_4, \mathbf{x}} \quad (15)$$

where  $\mathcal{C}_{p, p, \mathbf{s}}$ ,  $\mathcal{C}_{p, p, p, \mathbf{s}}$  and  $\mathcal{C}_{p, p, p, p, \mathbf{s}}$  denote the SO, TO and the FO marginal cumulants of the  $p$ -th source, respectively, and where  $R_{n_1, n_2, \mathbf{x}}$ ,  $R_{n_1, n_2, n_3, \mathbf{x}}$  and  $R_{n_1, n_2, n_3, n_4, \mathbf{x}}$  contains the terms depending on the non-zero cross-cumulants of the sources, due to their non-total independence. Note that the second-order cumulant of the Gaussian noise is included in the components  $R_{n_1, n_2, \mathbf{x}}$ .

We propose to merge together the entries of the cumulant arrays in a third-order array,  $\mathcal{X}$  of size  $(I \times I \times I^2 + I + 1)$ . In case of SO, TO and FO cumulants,

the  $(i_1, i_2, i_3)$ -th entry of  $\mathcal{X}$ , is given by:

$$\mathcal{X}_{i_1, i_2, i_3} = \begin{cases} \mathcal{C}_{i_1, i_2, n_3, n_4, \mathbf{x}} & \text{for any } (i_1, i_2) \in \{1, \dots, I\}^2 \\ & \text{and } i_3 \in \{1, \dots, I^2\} \text{ with } i_3 = n_3 + In_4. \\ \mathcal{C}_{i_1, i_2, i_3, \mathbf{x}} & \text{for any } (i_1, i_2) \in \{1, \dots, I\}^2 \text{ and} \\ & i_3 \in \{I^2+1, \dots, I^2+I\} \\ \mathcal{C}_{i_1, i_2, \mathbf{x}} & \text{for any } (i_1, i_2) \in \{1, \dots, N\}^2 \text{ and } i_3 = I^2+I+1 \end{cases} \quad (16)$$

The  $K$  frontal slices,  $\mathbf{X}^{(k)}$ , of  $\mathcal{X}$  follow the semi-nonnegative JDC model, with  $K = I^2 + I + 1$ :

$$\forall k \in \{1, 2, \dots, K\}, \mathbf{X}^{(k)} = \mathbf{A} \mathbf{D}^{(k)} \mathbf{A}^\top + \mathbf{R}^{(k)} \quad (17)$$

with:

$$\mathbf{D}^{(k)} = \begin{cases} \text{diag}(\mathbf{c}_{4, \mathbf{s}} \square (\mathbf{A} \odot \mathbf{A})_{k, :}) & \text{for any } k \in \{1, \dots, I^2\} \\ \text{diag}(\mathbf{c}_{3, \mathbf{s}} \square \mathbf{A}_{k-I^2, :}) & \text{for any } k \in \{I^2+1, \dots, I^2+I\} \\ \text{diag}(\mathbf{c}_{2, \mathbf{s}}) & \text{for } k = I^2+I+1 \end{cases} \quad (18)$$

with the  $P$ -dimensional vectors  $\mathbf{c}_{2, \mathbf{s}} = [\mathcal{C}_{1,1, \mathbf{s}}, \dots, \mathcal{C}_{P, P, \mathbf{s}}]$ ,  $\mathbf{c}_{3, \mathbf{s}} = [\mathcal{C}_{1,1,1, \mathbf{s}}, \dots, \mathcal{C}_{P, P, P, \mathbf{s}}]$  and  $\mathbf{c}_{4, \mathbf{s}} = [\mathcal{C}_{1,1,1,1, \mathbf{s}}, \dots, \mathcal{C}_{P, P, P, P, \mathbf{s}}]$ , and where  $\mathbf{A}_{i, :}$  (for  $1 \leq i \leq I$ ) and  $(\mathbf{A} \odot \mathbf{A})_{j, :}$  (for  $1 \leq j \leq I^2$ ) are the  $i$ -th row of  $\mathbf{A}$  and the  $j$ -th row of  $\mathbf{A} \odot \mathbf{A}$ , respectively. The semi-nonnegative ICA problem is then be reformulated as a semi-nonnegative JDC problem (17) and can be solved using the algorithms proposed in section 3.

#### 4. Simulations

The purpose of this section is twofold. First, it tackles the performance of the proposed methods, which exploit explicitly prior information, such as semi-nonnegativity and semi-symmetry. Second, the exponential change of variable is evaluated by comparing the performance of ELS-ALS<sup>exp</sup>, proposed in this paper, with the one of ELS-ALS<sup>square</sup> [24], based on the same optimization procedure and on a square parameterization to ensure the nonnegative constraints on  $\mathbf{A}$ . This analysis is performed in terms of accuracy of factor estimation and numerical complexity.

First, in section 4.2, the experiments are made on synthetic data, in order to evaluate the impact of the Signal to Noise Ratio (SNR) and the collinearity between columns of the nonnegative matrix factor,  $\mathbf{A}$ , on the behavior of the considered methods.  $\text{CG}^{\text{exp}}$  and  $\text{ELS-ALS}^{\text{exp}}$  are compared with i) two classical CP algorithms, called ELS-ALS [27] and LM [3], where neither symmetry nor non-negativity constraints are imposed, with ii) the Alternating Columns/Diagonal Center (ACDC) algorithm [7], which solves the JDC problem, taken into account the symmetry constraint and with iii) a semi-nonnegative JDC algorithm, based on a square change of variable  $\text{ELS-ALS}^{\text{square}}$  [24]. Second, in section 4.3, the behavior of the methods proposed in section 3.5 is assessed in ICA applications w.r.t. a classical Nonnegative Matrix Factorization (NMF) algorithm called NNLS [29] and four classical ICA methods, namely SOBI [8], FastICA [36], eJade(3,4) [37] and CoM2 [38]. Experiments are carried out on simulated MRS data, in section 4.3.1, and on image data, in section 4.3.2.

#### 4.1. Performance criterion

As a performance criterion, we use a global measure allowing us to quantify the error between the actual factor matrices,  $\mathbf{A}$  and  $\mathbf{D}^{(k)}$  and their estimates, named  $\hat{\mathbf{A}}$  and  $\hat{\mathbf{D}}^{(k)}$ , respectively. Note that if  $\hat{\mathbf{A}}$  and  $\hat{\mathbf{D}}^{(k)}$  are solutions of semi-nonnegative JDC problem, then it is also true for any matrix couple of the form  $(\hat{\mathbf{A}}\mathbf{\Lambda}\mathbf{P}, \mathbf{P}^\top\mathbf{\Lambda}^{-1}\hat{\mathbf{D}}^{(k)}\mathbf{\Lambda}^{-1}\mathbf{P})$ , where  $\mathbf{\Lambda}$  and  $\mathbf{P}$  are a diagonal matrix and a permutation matrix, respectively. So, our semi-nonnegative JDC problem is solved up to scaling and permutation indeterminacies. In this context, the proposed measure must be invariant under both matrices,  $\mathbf{\Lambda}$  and  $\mathbf{P}$ . For  $\mathbf{A}$  and all matrices  $\mathbf{D}^{(k)}$ , the performance criterion, called  $\alpha$ , is given by:

$$\alpha = \frac{1}{2P} \sum_{k=1}^P \Delta_k^A + \Delta_k^C \quad (19)$$

with  $\mathbf{C} = [\text{diag}(\mathbf{D}^{(1)}), \text{diag}(\mathbf{D}^{(2)}), \dots, \text{diag}(\mathbf{D}^{(K)})]^\top$  and where:

$$\Delta_k^G = \min_{(p,p') \in M_k^2} d(\mathbf{g}_p, \hat{\mathbf{g}}_{p'}) \quad (20)$$



with  $\mathbf{g}_p$  and  $\hat{\mathbf{g}}_p$  the  $p$ -th column of  $\mathbf{G}$  and  $\hat{\mathbf{G}}$ , respectively, and  $M^2 = \{1, \dots, P\} \times \{1, \dots, P\}$ . Besides,  $d$  is the pseudo-distance between the vectors  $\mathbf{u}$  and  $\mathbf{v}$  defined by [39]:

$$d(\mathbf{u}, \mathbf{v}) = 1 - \frac{\|\mathbf{u}^\top \mathbf{v}\|^2}{\|\mathbf{u}\|^2 \|\mathbf{v}\|^2} \quad (21)$$

In the first step aiming at computing  $\alpha$ , the smallest distance is selected, associating a column of a loading matrix with one of its estimate. Then, this pair of indices is removed from  $M^2$ , and the step is repeated until the  $P$  columns of the estimate are used and the set  $M^2$  is empty. So, each column of  $\mathbf{G}$  is associated with a different column  $\hat{\mathbf{G}}$  and the measure  $\alpha$  is the sum of all distances  $d$  between each pair of columns. This criterion is an upper bound of the optimal criterion [28]. It avoids the computation of all permutations between the columns of the estimated and actual factor matrices, in order to use it in the context of arrays with high dimensions and rank.

#### 4.2. Synthetic data

Synthetic sets of matrices following the semi-nonnegative JDC model are generated from random matrices. In particular, the matrix  $\mathbf{E}$ , used in  $\mathbf{A} = \exp(\mathbf{E})$ , is simulated with the zero-mean unit-variance normal distribution and the diagonals of the matrices belonging to the set  $\{\mathbf{D}^{(1)}, \dots, \mathbf{D}^{(K)}\}$  are dropped from a zero-mean Gaussian distribution with unit standard deviation. Moreover, each resulting matrix,  $\mathbf{X}^{(k)}$ , is contaminated by a noise, named  $\mathbf{R}^{(k)}$ , following a semi-nonnegative JDC model, generated like  $\mathbf{X}^{(k)}$ . Each corresponding noisy matrix  $\mathbf{Y}^{(k)}$  is written as:

$$\mathbf{Y}^{(k)} = \frac{\mathbf{X}^{(k)}}{\|\mathbf{X}^{(k)}\|_F} + \sigma_R \frac{\mathbf{R}^{(k)}}{\|\mathbf{R}^{(k)}\|_F} \quad (22)$$

where  $\sigma_R$  is a scalar controlling the noise level. The Signal to Noise Ratio (SNR) is then defined by:  $\text{SNR} = -20 \log_{10}(\sigma_R)$ .

All algorithms considered in this study stop either when the relative error of the cost function between two successive iterations exhibits a value less than a predefined threshold of  $10^{-8}$  or when the number of iterations exceeds 2000.

Furthermore, all results reported in this section are averaged over 200 Monte Carlo realizations. In addition, we used a multi-start random initialization of  $\hat{\mathbf{E}}$  and  $\hat{\mathbf{C}}$ , dropped from a zero-mean unit-variance normal distributions.

In this experiment, the impact of SNR on the behavior of the considered methods is evaluated. Both the dimensions and the rank of  $\mathcal{T}$  are fixed to  $(5 \times 5 \times 5)$  and 3, respectively. Figure 1(a) shows the mean of the measure  $\alpha$  at the output of all methods as a function of the SNR ranging for -20 dB to 60 dB. As expected, the decomposition is more effective in the context of high SNR values. Both proposed methods,  $\text{CG}^{\text{exp}}$  and  $\text{ELS-ALS}^{\text{exp}}$  outperform the CP and JDC methods, namely ELS-ALS, LM and ACDC. Our results in terms of estimation accuracy are similar to the semi-nonnegative JDC algorithm based on a square change of variable, namely  $\text{ELS-ALS}^{\text{square}}$ . Figure 1(b) displays the mean of numerical complexity for each method as a function of the SNR. Among our methods,  $\text{ELS-ALS}^{\text{exp}}$  is the cheapest for every SNR values and its cost is even less expensive to that of the unconstrained ELS-ALS methods. Regarding  $\text{CG}^{\text{exp}}$  and  $\text{ELS-ALS}^{\text{square}}$ , in spite of good similar estimation accuracy, their numerical complexities are higher. In this simulation, we note the benefit of exploiting an exponential change of variable, in terms of numerical complexity. This can straightforwardly be seen by comparing the performance of the ELS-ALS-like algorithm based on a square change of variable,  $\text{ELS-ALS}^{\text{square}}$ , with its exponential version, i.e. the  $\text{ELS-ALS}^{\text{exp}}$  algorithm. So, the best accuracy/complexity trade off is achieved by  $\text{ELS-ALS}^{\text{exp}}$ .

In a second simulation, the impact of the collinearity between columns of the nonnegative matrix factor  $\mathbf{A}$  is evaluated. A set of  $(10 \times 4)$  matrices  $\mathbf{A}$  is generated with two bottlenecks such as  $\mathbf{A} = [\mathbf{a}_1; \mathbf{a}_1 + \beta \mathbf{a}_2; \mathbf{a}_3; \mathbf{a}_3 + \beta \mathbf{a}_4]$ , where  $\mathbf{a}_i$  is drawn from a standard uniform distribution on  $[0, 1]$ , for  $1 \leq i \leq 4$ , and  $\beta$  ranging from 0 to 1 [28]. For an SNR value equal to 60dB, figures 2(a) and 2(b) show the mean of the measure  $\alpha$  and the numerical complexity at the output of all methods as a function of the parameter  $\beta$ , respectively. When vectors belonging to a bottleneck are close to collinear ( $\beta \approx 0$ ), the context of the simulation is very difficult and the decomposition is less effective in terms

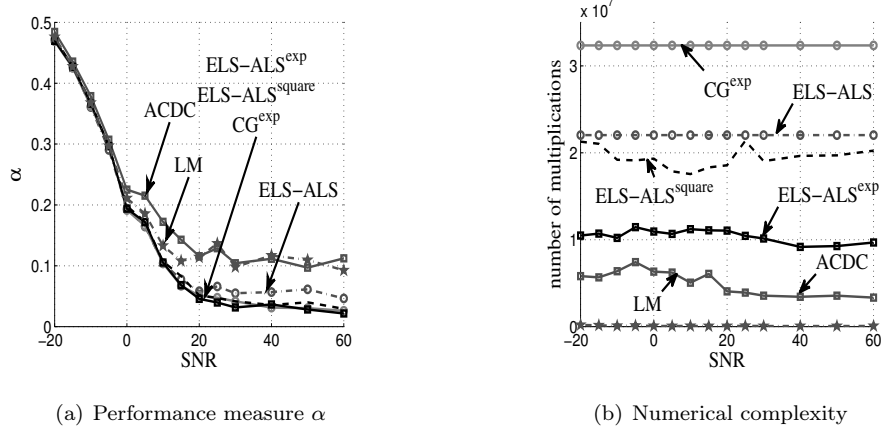


Figure 1: Influence of SNR for a  $(5 \times 5 \times 5)$  array and  $P = 3$  at the output of both proposed methods, called  $\text{CG}^{\text{exp}}$  and  $\text{ELS-ALS}^{\text{exp}}$ , a semi-nonnegative JDC algorithm,  $\text{ELS-ALS}^{\text{square}}$ , a JDC method, ACDC, and two CP methods, namely ELS-ALS and LM.

of estimation accuracy. As shown in figure 2(a), the measures  $\alpha$  of all methods decrease to zero at various speeds. The semi-nonnegative JDC methods are less sensitive to high collinearity coefficients. Indeed, contrary to the CP and JDC methods, the semi-nonnegative JDC methods are effective in terms of estimation accuracy for  $\beta$  beyond 0.2.  $\text{ELS-ALS}^{\text{exp}}$  is the least expensive and it is even less costly beyond to  $\beta = 0.4$  than unconstrained ELS-ALS algorithm. So, in terms of numerical complexity,  $\text{ELS-ALS}^{\text{exp}}$  is favored between the three algorithms giving the best estimation accuracy, namely  $\text{CG}^{\text{exp}}$ ,  $\text{ELS-ALS}^{\text{exp}}$  and  $\text{ELS-ALS}^{\text{square}}$ .

#### 4.3. Semi-nonnegative ICA applications

The goal of this section is to illustrate the semi-nonnegative ICA problem and to assess the behavior of two semi-nonnegative ICA methods, proposed in this paper.

##### 4.3.1. Tests on simulated MRS data

In this section, the semi-nonnegative ICA methods are tested on simulated MRS data. In such a context, as explained in section 3.5, the metabolite spec-

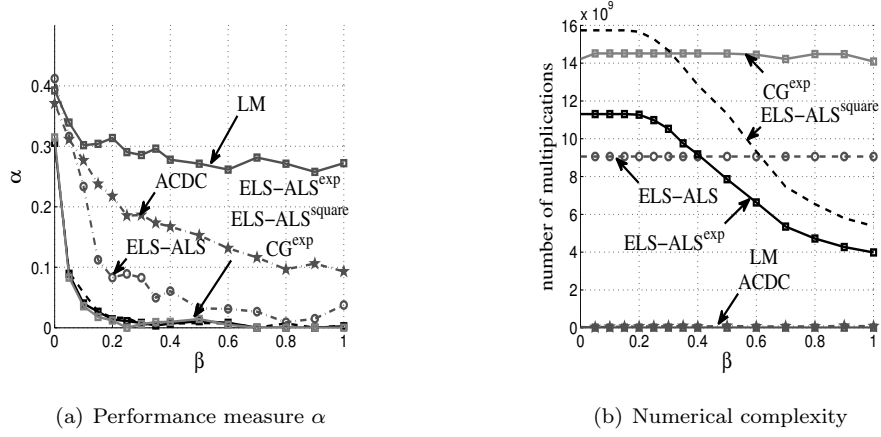


Figure 2: Influence of the collinearity for a  $(10 \times 10 \times 10)$  array and  $P = 4$  at the output of both proposed methods,  $\text{CG}^{\text{exp}}$  and  $\text{ELS-ALS}^{\text{exp}}$ , a semi-nonnegative JDC algorithm,  $\text{ELS-ALS}_{\text{square}}$ , a JDC method, ACDC, and two CP methods, ELS-ALS and LM.

tra are considered not totally independent. After studying the behavior of the cumulants, only the FO cumulant array is used in our semi-nonnegative ICA algorithms. This can straightforwardly be seen by comparing the ratio between the highest cross-cumulant and the lowest marginal cumulant of metabolite spectra for the SO, TO and FO cumulants. In this context, this ratio is higher when considering FO cumulant. Indeed, for a mixture of choline and myo-inositol, we obtained 0.03, 0.16 and 0.53 with the fourth order, third order and second order cumulants, respectively. So, we suppose that exploiting only the FO cumulants gives better results. This report was been proved by comparing the performance of the two semi-nonnegative algorithms with their versions exploiting the SO, TO and FO cumulants. In addition, the concentrations of the metabolites are positive, ensuring the nonnegativity of the mixing matrix  $\mathbf{A}$ . Several metabolite spectra are generated by the sum of Lorentzian and Gaussian functions where the location and scale parameters are fixed to derive realistic metabolites [35]. The coefficients of the mixing matrix  $\mathbf{A}$  are drawn from a uniform distribution on  $[0, 1]$ . The number of samples is 2500. The observations are generated as a noisy mixture of metabolite sources. As far as the additive noise is considered,

Algorithms	$I$				
	2	4	8	12	
SOBI	0.303	0.082	0.080	0.080	
CoM2	0.095	0.037	0.033	0.033	
eJade(3,4)	0.110	0.042	0.037	0.035	
FastICA	0.095	0.039	0.035	0.035	
NNLS	0.161	0.089	0.036	0.030	
ELS-ALS <sup>exp</sup>	0.116	0.017	0.007	0.007	
CG <sup>exp</sup>	0.089	0.012	0.007	0.007	

Table 1: Simulation 1 - choline/myo-inositol- Performance measure  $\gamma$  with respect to different values of  $I$  at the output of four classical ICA approaches, called SOBI, CoM2, eJade(3,4) and FastICA, a classical NMF technique, NNLS, and the two proposed methods, ELS-ALS<sup>exp</sup> and CG<sup>exp</sup>, for an SNR value of 30dB.

a normal vector process is used to simulate a realistic instrumental noise and the SNR value is equal to 30dB. As a performance criterion, we use the global measure  $\gamma$  allowing to quantify the estimation error of the sources, as follows:

$$\gamma = \frac{1}{2P} \sum_{k=1}^P \Delta_k^S \quad (23)$$

with  $\Delta_k^S$  given in (20).

The impact of the number of observations is evaluated. Two mixtures of two and three sources of interest, namely choline/myo-inositol and choline/myo-inositol/N-acetyl Aspartate, are considered. Tables 1 and 3 show the mean of the measure  $\gamma$  of the sources, for the two mixtures, at the output of the seven methods as a function of the number  $I$  of observations. Three ICA algorithms, called CoM2, eJade(3,4) and FastICA, have similar results and outperform the NMF algorithm, NNLS, and the last ICA approach, SOBI, for  $I \in \{4, 8, 12\}$ . Beyond  $I = 4$ , the performance of our methods increases with the value of  $I$ , contrary to ICA algorithms, having a constant behavior. Our methods outperform the classical ICA and NMF methods as soon as the number of observations  $I$  increases beyond  $I = 4$  and  $I = 8$  for the first and second simulations, respec-

Algorithms \ $I$	2	4	8	12
SOBI	$0.15 \times 10^5$	$0.40 \times 10^5$	$1.20 \times 10^5$	$2.42 \times 10^5$
CoM2	$0.63 \times 10^5$	$0.88 \times 10^5$	$1.69 \times 10^5$	$2.90 \times 10^5$
eJade(3,4)	$0.85 \times 10^5$	$1.09 \times 10^5$	$1.90 \times 10^5$	$3.12 \times 10^5$
FastICA	$0.05 \times 10^7$	$0.15 \times 10^7$	$0.55 \times 10^7$	$1.32 \times 10^7$
NNLS	$0.01 \times 10^{10}$	$0.08 \times 10^{10}$	$0.83 \times 10^{10}$	$3.70 \times 10^{10}$
ELS-ALS <sup>exp</sup>	$4.86 \times 10^8$	$3.97 \times 10^8$	$3.32 \times 10^8$	$4.75 \times 10^8$
CG <sup>exp</sup>	$5.52 \times 10^8$	$2.32 \times 10^8$	$0.54 \times 10^8$	$0.76 \times 10^8$

Table 2: Simulation 1 - choline/myo-inositol- Numerical complexity with respect to different values of  $I$  at the output of four classical ICA approaches, called SOBI, CoM2, eJade(3,4) and FastICA, a classical NMF technique, NNLS, and the two proposed methods, ELS-ALS<sup>exp</sup> and CG<sup>exp</sup>, for an SNR value of 30dB.

tively. Concerning our semi-nonnegative JDC algorithms, their behaviors are similar in terms of  $\gamma$ . Tables 2 and 4 display the mean of numerical complexity for each method as a function of the number of observations for the two scenarios. As expected, the numerical complexity increases with  $I$ . The classical ICA methods are less expensive than our semi-nonnegative ICA methods and NMF. Our proposed methods require at least 100 times more flops than SOBI, eJade(3,4) and CoM2. NNLS has the highest numerical complexity.

For the first simulation, figure 3 shows the mean over 200 realizations of the estimated sources at the output of the seven methods for  $I = 12$ . For the different algorithms, the values of  $\gamma$  are 0.080 for SOBI, 0.033 for CoM2, 0.035 for FastICA, 0.035 for eJade(3,4), 0.030 for NNLS, and 0.007 for our two semi-nonnegative JDC methods. For NNLS, Myo-Inositol is still present in the first extracted source. Concerning SOBI, CoM2, eJade(3,4) and FastICA, the shapes of their first sources are distorted reducing the area under the curve. Their estimated source has negative components. As far as the semi-nonnegative methods are concerned, the separation of both metabolites is perfect. This experiment shows that in this context the use of both constraints, namely nonnegativity of

Algorithms	$I$	3	4	8	12
SOBI		0.373	0.206	0.102	0.102
CoM2		0.139	0.062	0.041	0.036
eJade(3,4)		0.139	0.049	0.029	0.028
FastICA		0.136	0.058	0.031	0.030
NNLS		0.204	0.210	0.108	0.101
ELS-ALS <sup>exp</sup>		0.202	0.084	0.013	0.010
CG <sup>exp</sup>		0.204	0.109	0.012	0.010

Table 3: Simulation 2 - choline/Myo-Inositol/N-acetylAspartate - Performance measure  $\gamma$  with respect to different values of  $I$  at the output of four classical ICA approaches, called SOBI, CoM2, eJade(3,4) and FastICA, a classical NMF technique, NNLS, and the two proposed methods, ELS-ALS<sup>exp</sup> and CG<sup>exp</sup>, for an SNR value of 30dB.

the mixing matrix and quasi-independence of the sources, allows us to achieve good estimation accuracy.

#### 4.3.2. Tests on image data

In this section, the semi-nonnegative ICA problem is illustrated through a BSS image processing application. A database of 40 face images [40] is considered in order to generate 780 couples of sources. The nonnegative mixing matrix  $\mathbf{A}$  is drawn from a uniform distribution on  $[0, 1]$ . A Gaussian vector process is added to each mixture to simulate a realistic instrumental noise and the SNR value is equal to 60dB. Table 5 shows the mean over the 780 realizations of the performance criterion  $\gamma$  of sources and the numerical complexity for all datasets at the output of the seven considered methods. The measure of the NMF method gives better results than the ones of the four classical ICA methods. However, both semi-nonnegative ICA methods, ELS-ALS<sup>exp</sup> and CG<sup>exp</sup>, outperform ICA and NMF algorithms. This experiment shows that in this context the use of both constraints, namely nonnegativity of the mixing matrix and quasi-independence of the sources, allows us again to achieve better estimation

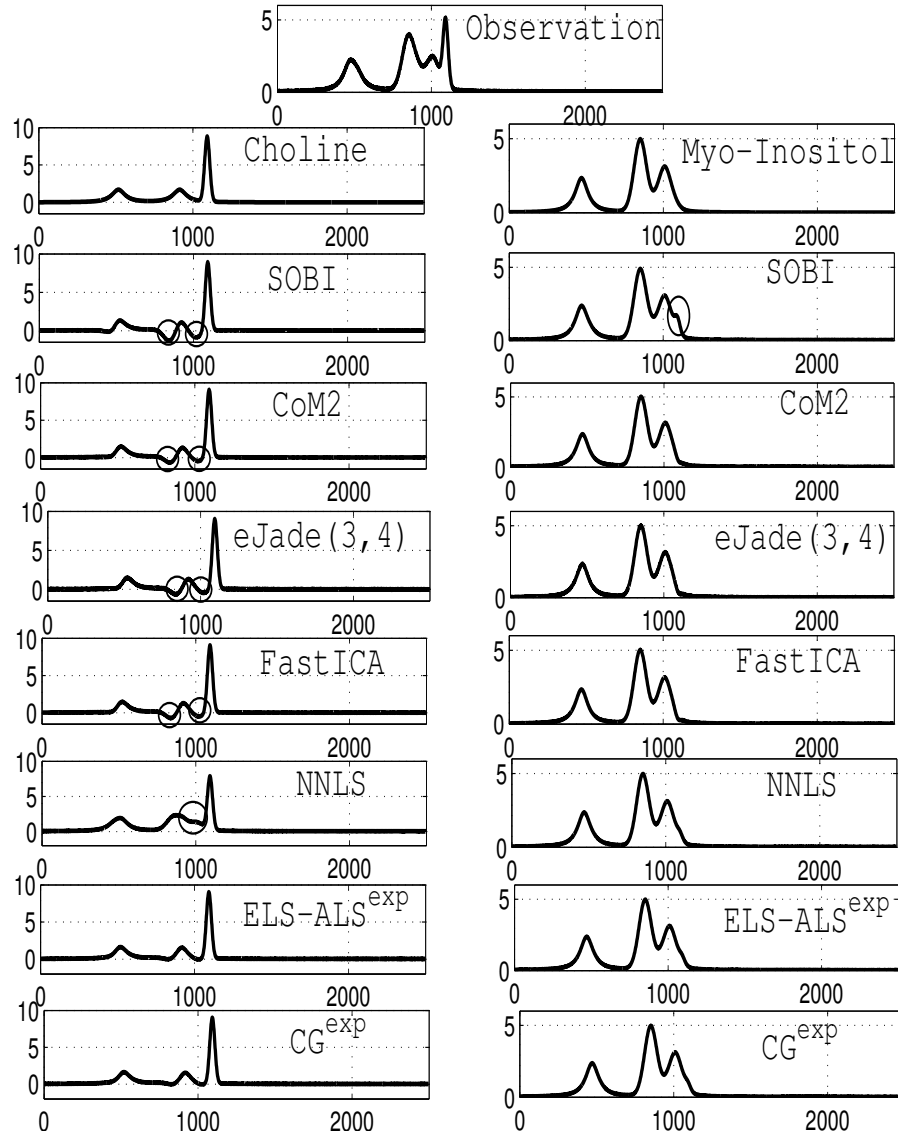


Figure 3: Mean over 200 realizations of two sources, choline and myo-inositol, extracted by SOBI, CoM2, eJade(3,4) FastICA, NNLS and our two semi-nonnegative ICA methods, based on ELS-ALS<sup>exp</sup> and CG<sup>exp</sup>, for an SNR value of 30dB.



Algorithms	$I$	3	4	8	12
SOBI		$0.35 \times 10^5$	$0.51 \times 10^5$	$1.42 \times 10^5$	$2.73 \times 10^5$
CoM2		$1.74 \times 10^5$	$1.90 \times 10^5$	$1.82 \times 10^5$	$4.14 \times 10^5$
eJade(3,4)		$2.66 \times 10^5$	$2.82 \times 10^5$	$3.73 \times 10^5$	$6.30 \times 10^5$
FastICA		$0.19 \times 10^7$	$0.31 \times 10^7$	$1.32 \times 10^7$	$2.42 \times 10^7$
NNLS		$0.01 \times 10^{10}$	$0.08 \times 10^{10}$	$0.83 \times 10^{10}$	$3.70 \times 10^{10}$
ELS-ALS <sup>exp</sup>		$6.16 \times 10^8$	$4.90 \times 10^8$	$1.66 \times 10^8$	$4.93 \times 10^8$
CG <sup>exp</sup>		$1.00 \times 10^8$	$0.96 \times 10^8$	$1.57 \times 10^8$	$3.03 \times 10^8$

Table 4: Simulation 2 - choline/Myo-Inositol/N-acetylAspartate - Numerical complexity with respect to different values of  $I$  at the output of four classical ICA approaches, called SOBI, CoM2, eJade(3,4) and FastICA, a classical NMF technique, NNLS, and the two proposed methods, ELS-ALS<sup>exp</sup> and CG<sup>exp</sup>, for an SNR value of 30dB.

accuracy. Concerning the numerical complexity, SOBI is the least expensive among all methods. NNLS has the highest cost a long way ahead. Among the semi-nonnegative ICA techniques, ELS-ALS<sup>exp</sup> is less expensive (similar to CoM2) and it may give a good compromise between accuracy and complexity.

Figure 4 shows the extracted images at the output of the seven methods, for three realizations. The first and second lines correspond to the true images and the corresponding mixtures, respectively. The four ICA methods, SOBI, CoM2, eJade(3,4) and FastICA, fail to separate both images. Often only one image is properly extracted and the second one is a mixture of the two face images. On the other hand, ELS-ALS<sup>exp</sup> and CG<sup>exp</sup> allow us to achieve a good image extraction.

## 5. Conclusion

In order to achieve the semi-nonnegative JDC decomposition, we have proposed two algorithms named ELS-ALS<sup>exp</sup> and CG<sup>exp</sup>. The exponential change of variable is used to ensure the nonnegativity of the joint diagonalizer, leading to an unconstrained optimization problem. A line search procedure has been

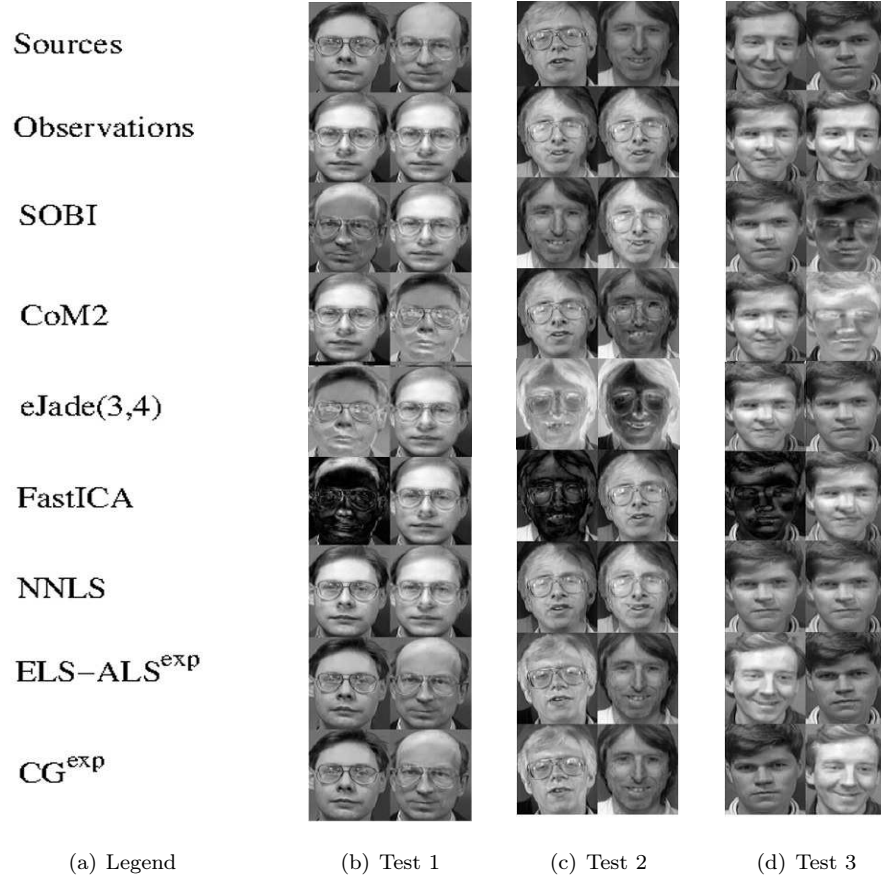


Figure 4: Results for 3 realizations: (line 1) the true images, (line 2) the corresponding mixtures, (line 3) the extracted images by SOBI, (line 4) by CoM2,(line 5) by eJade(3,4), (line 6) by FastICA, (line 7) by NNLS, (line 8) by ELS-ALS<sup>exp</sup> and (line 9) by CG<sup>exp</sup>.

Algorithms	measure $\alpha$	number of multiplications
SOBI	0.286	$5.52 \times 10^3$
CoM2	0.109	$2.47 \times 10^6$
eJade(3,4)	0.125	$4.42 \times 10^6$
FastICA	0.30	$5.83 \times 10^7$
NNLS	0.112	$3.09 \times 10^{11}$
ELS-ALS <sup>exp</sup>	0.0374	$4.33 \times 10^6$
CG <sup>exp</sup>	0.0377	$7.22 \times 10^7$

Table 5: Average values of measure  $\gamma$  and numerical complexity at the output of four classical ICA methods (SOBI, CoM2, eJade(3,4) and FastICA), one NMF algorithm (NNLS) and the proposed algorithms, called ELS-ALS<sup>exp</sup> and CG<sup>exp</sup>.

considered, in order to accelerate the convergence. The derivatives used in the conjugate gradient-like method and in the ALS step of ELS-ALS<sup>exp</sup> have been jointly calculated in a matrix form using the algebraic basis for matrix calculus and product operator properties. Besides, the computational complexity has been evaluated analytically for each algorithm.

In section 4, our semi-nonnegative JDC algorithms are tested on synthetic data, to evaluate the performance of the methods and the interest of using jointly both *a priori* (*i.e.* semi-nonnegativity and semi-symmetry). The classical ELS-ALS [27] and LM [3] algorithms without symmetry and nonnegativity constraints, and the ACDC algorithm [7] where only the semi-symmetry constraint is imposed, are tested as reference methods. Furthermore, the performance is also compared with a semi-nonnegative JDC algorithm based on a square change of variable and a ELS-ALS-like optimization method [24]. The analysis is performed in terms of accuracy of factor estimation and numerical complexity. The comparison studies show that a better estimation accuracy is obtained when these two *a priori* are exploited. Among these approaches, the best accuracy/complexity trade off is achieved by ELS-ALS<sup>exp</sup>, thanks to its faster convergence speed.

The semi-nonnegative JDC analysis turns up in some ICA applications, which involve nonnegative static mixtures, such as image processing or MRS. Indeed, it allows us to solve what we called the semi-nonnegative ICA problem, imposing a nonnegative constraint on the mixing matrix. We propose cumulant-based methods to blindly identify potentially underdetermined mixtures of statistically independent sources. We illustrate the behavior of our semi-nonnegative ICA techniques in two BSS applications. Some experiments are carried out on simulated MRS spectra and on image data. Our methods are compared to a classical NMF algorithm, called NNLS and four classical ICA methods, named SOBI, CoM2, eJade(3,4) and FastICA. Those experiments show that the use of both constraints, namely nonnegativity of the mixing matrix and quasi-independence of the sources, allows us to achieve better estimation accuracy.

## Appendix A. Gradient computation

The first-order derivative of the objective function  $\Psi$  (2) is calculated in a compact matrix form [41]. From equation (2), we have:

$$\Psi(\mathbf{E}, \mathbf{C}) = \text{Tr}(\mathbf{X}_3^\top \mathbf{X}_3) - 2f(\mathbf{E}, \mathbf{C}) + g(\mathbf{E}, \mathbf{C}) \quad (\text{A.1})$$

with  $f(\mathbf{E}, \mathbf{C}) = \text{Tr}(\mathbf{X}_3^\top \mathbf{C} \mathbf{Z}^\top)$ ,  $g(\mathbf{E}, \mathbf{C}) = \mathbf{1}_P^\top ((\mathbf{A}^\top \mathbf{A})^{\square 2} \square \mathbf{C}^\top \mathbf{C}) \mathbf{1}_P$ ,  $\mathbf{Z} = \mathbf{A} \odot \mathbf{A}$  and  $\mathbf{A} = \exp(\mathbf{E})$ . The differential of  $\Psi$ , seen as a scalar function of two real-valued matrices  $\mathbf{E}$  and  $\mathbf{C}$ , is given by:

$$d\Psi(\mathbf{E}, \mathbf{C}) = D_{\mathbf{E}}\Psi(\mathbf{E}, \mathbf{C}) \text{dvec}(\mathbf{E}) + D_{\mathbf{C}}\Psi(\mathbf{E}, \mathbf{C}) \text{dvec}(\mathbf{C})$$

where  $D_{\mathbf{E}}\Psi(\mathbf{E}, \mathbf{C}) = \partial\Psi/\partial\text{vec}(\mathbf{E})^\top$  and  $D_{\mathbf{C}}\Psi(\mathbf{E}, \mathbf{C}) = \partial\Psi/\partial\text{vec}(\mathbf{C})^\top$  represent the partial derivatives of  $\Psi$  with respect to (w.r.t.)  $\mathbf{E}$  and  $\mathbf{C}$ , respectively. Then, by using the chain rules and differential properties, the following two equations are obtained:

$$D_{\mathbf{E}}\Psi(\mathbf{E}, \mathbf{C}) = D_{\mathbf{Z}}f(\mathbf{E}, \mathbf{C}) D_{\mathbf{A}}\mathbf{Z} D_{\mathbf{E}}\mathbf{A} + D_{\mathbf{Z}}g(\mathbf{E}, \mathbf{C}) D_{\mathbf{A}}\mathbf{Z} D_{\mathbf{E}}\mathbf{A} \quad (\text{A.2})$$

$$D_{\mathbf{C}}\Psi(\mathbf{E}, \mathbf{C}) = D_{\mathbf{C}}f(\mathbf{E}, \mathbf{C}) + D_{\mathbf{C}}g(\mathbf{E}, \mathbf{C}) \quad (\text{A.3})$$

Note that  $D_{\mathbf{A}}\mathbf{Z}$  and  $D_{\mathbf{B}}\mathbf{A}$  are defined as:

$$D_{\mathbf{A}}\mathbf{Z} = \partial \text{vec}(\mathbf{Z}) / \partial \text{vec}(\mathbf{A})^\top = \text{diag}(\text{vec}(\mathbf{1}_I \otimes \mathbf{A}))(\mathbf{I}_{IP} \otimes \mathbf{1}_I) + \quad (\text{A.4})$$

$$\text{diag}(\text{vec}(\mathbf{A}) \otimes \mathbf{1}_I)(\mathbf{U}_{IP} \otimes \mathbf{I}_I)(\mathbf{1}_I \otimes \mathbf{I}_{IP})$$

$$D_{\mathbf{E}}\mathbf{A} = \partial \text{vec}(\mathbf{A}) / \partial \text{vec}(\mathbf{E})^\top = \text{diag}(\text{vec}(\exp(\mathbf{E}))) \quad (\text{A.5})$$

Then, we get, using the expressions of  $D_{\mathbf{A}}\mathbf{Z}$  and  $D_{\mathbf{E}}\mathbf{A}$ :

**Lemma 1.** *The gradient  $D\Psi(\mathbf{E}, \mathbf{C})$  is given by:*

$$D\Psi(\mathbf{E}, \mathbf{C}) = [D_{\mathbf{E}}\Psi(\mathbf{E}, \mathbf{C}), D_{\mathbf{C}}\Psi(\mathbf{E}, \mathbf{C})] \quad (\text{A.6})$$

with:

$$\begin{aligned} D_{\mathbf{E}}\Psi(\mathbf{E}, \mathbf{C}) = & -2(\text{vec}(\exp(\mathbf{E}))\boxtimes(((\mathbf{A} \otimes \mathbf{1}_I)^\top \boxtimes (\mathbf{M} + \mathbf{N}))^\top \mathbf{1}_I))^\top \\ & + 2\text{vec}(\mathbf{A}^\top \mathbf{A} \boxtimes \mathbf{C}^\top \mathbf{C})^\top ((\mathbf{I}_P \otimes \mathbf{A}^\top) + (\mathbf{A}^\top \otimes \mathbf{I}_P)\mathbf{U}_{IP})\text{diag}(\text{vec}(\exp(\mathbf{E}))) \end{aligned}$$

and:

$$D_{\mathbf{C}}\Psi(\mathbf{E}, \mathbf{C}) = -2\text{vec}(\mathbf{X}_3 \mathbf{Z})^\top + 2\text{vec}(\mathbf{C})^\top (\mathbf{Z}^\top \mathbf{Z} \otimes \mathbf{I}_K)$$

with  $\mathbf{A} = \exp(\mathbf{E})$ ,  $\mathbf{Z} = \mathbf{A} \odot \mathbf{A}$ ,  $\mathbf{M} = \text{Mat}^{(I \times 1, IP)}(\text{vec}(\mathbf{X}_3^\top \mathbf{C}))$  and  $\mathbf{N} = \text{Mat}^{(I \times I, P)}(\mathbf{M}^\top)$ .

## Appendix B. ALS computation

In order to minimize the subproblem (6) w.r.t. the  $(a, b)$ -th component of  $\mathbf{E}$ , we introduce  $\Phi_{E_{a,b}} : E_{a,b} \mapsto \Psi(\mathbf{E}, \mathbf{C})$ , corresponding to the function  $\Psi$  w.r.t.  $E_{a,b}$ , while keeping the other elements of  $\mathbf{E}$  fixed. We rewrite  $\Phi_{E_{a,b}}$  as a composite function defined by  $\Phi_{E_{a,b}}(E_{a,b}) = g \circ f(E_{a,b})$  with  $f : E_{a,b} \mapsto \exp(E_{a,b})$  and  $g : A_{a,b} \mapsto \left\| \mathbf{X}_3 - \mathbf{C} ((\mathbf{A})^{\odot 2})^\top \right\|_F^2$ . Function  $f$  is associated to the exponential change of variable and  $g$  corresponds to the JDC cost function w.r.t. the  $(a, b)$ -th component of  $\mathbf{A}$ . Consequently, we have:

$$g(A_{a,b}) = \beta_4(A_{a,b})^4 + \beta_2(A_{a,b})^2 + \beta_1 A_{a,b} + \beta_0 \quad (\text{B.1})$$

where:

$$\begin{aligned}
\beta_1 &= -2(\mathbf{A}^\top(\mathbf{M} + \mathbf{N}))_{b,(b-1)I+a} - 4A_{a,b}((\mathbf{C}^\top \mathbf{C}(\mathbf{A}^\top)^{\square 2}) \\
&\quad + (\mathbf{A}^\top \mathbf{A})_{b,b}(\mathbf{C}^\top \mathbf{C})_{b,b} + (A_{a,b})^3(\mathbf{C}^\top \mathbf{C})_{b,b}) \\
&\quad - 2A_{a,b}(\mathbf{M} + \mathbf{N})_{a,(b-1)I+a} \\
\beta_2 &= -2(\mathbf{M} + \mathbf{N})_{a,(b-1)I+a} + 4(\mathbf{A}^{\square 2} \mathbf{C}^\top \mathbf{C})_{a,b} \\
&\quad + 4((\mathbf{A}^\top \mathbf{A})_{b,b}(\mathbf{C}^\top \mathbf{C})_{b,b}) \\
\beta_4 &= 4(\mathbf{C}^\top \mathbf{C})_{b,b}
\end{aligned}$$

Note that  $\beta_0$  is not defined here, because this expression is very complicated and it is not useful in the sequel. A matrix computation of the coefficients  $\beta_i$ , for  $i \in \{1, 2, 4\}$ , is performed allowing for a straightforward implementation in matrix programming environments.

According to the derivative property of the composite function, the derivative of  $\Phi_{E_{a,b}}$ , denoted  $\Phi'_{E_{a,b}}$ , is given by:

$$\Phi'_{E_{a,b}}(E_{a,b}) = g'(f(E_{a,b}))f'(E_{a,b}) \quad (\text{B.2})$$

$$= g'(A_{a,b})f'(E_{a,b}) \quad (\text{B.3})$$

with:

$$f'(E_{a,b}) = \exp(E_{a,b}) \quad (\text{B.4})$$

$$g'(A_{a,b}) = 4\beta_4(A_{a,b})^3 + 2\beta_2A_{a,b} + \beta_1 \quad (\text{B.5})$$

As a result, we can search for the stationary points of  $\Phi'_{E_{a,b}}$  by computing the strictly positive roots of a third degree polynomial  $g'(A_{a,b})$  of the form  $h(Z) = Z^3 + pZ + q$ , with  $p = \beta_2/\beta_4$  and  $q = \beta_1/\beta_4$ . The analytical root can be computed using Cardano's method [42], which consists first in computing a discriminant  $\Delta$ . When  $\Delta \leq 0$ , one of the solutions is positive and we are done. On the other hand, when  $\Delta$  is strictly positive, the unique solution may be negative. We studied in detail the behavior of the polynomial function  $h$  in this particular case, and determined that the function  $g$  is strictly increasing on  $]0, +\infty[$ . Indeed,  $g$  cannot be defined in zero, because it depends on  $A_{a,b} = E_{a,b}$ .

So, the minimum of the cost function is achieved when the component of  $\mathbf{A}$  is equal to a small positive value  $\epsilon$  close to zero. The variable  $\epsilon$  decreases to zero as a function of the number of iterations. Indeed, for the first ten iterations, its is set to 0.1. Then, this value is divided by ten every ten iterations. At the end of the convergence, this procedure permits us to fix the minimum to zero, although this value is not allowed by the exponential change of variable, and thus to process mixtures with some exact zeros. Paatero used the same technique for his nonnegative tensor factorization algorithms, for which the strictly positivity of the component is ensured by a logarithmic penalty function [43].

### Appendix C. Calculation of $\varphi$

Note that in this section, the superscript  $^{it}$  will be omitted.  $\varphi$  (11) can be rewritten as follows:

$$\begin{aligned} \varphi(\mu_E, \mu_C) \approx & \left\| \mathbf{F}_0 + \mathbf{F}_1 \mu_E + \mathbf{F}_2 \mu_E^2 + \mathbf{F}_3 \mu_E^3 + \mathbf{F}_4 \mu_E^4 \right. \\ & + \mathbf{F}_5 \mu_E^5 + \mathbf{F}_6 \mu_E^6 + \mathbf{F}_7 \mu_C + \mathbf{F}_8 \mu_C \mu_E + \mathbf{F}_9 \mu_C \mu_E^2 + \\ & \left. \mathbf{F}_{10} \mu_C \mu_E^3 + \mathbf{F}_{11} \mu_C \mu_E^4 + \mathbf{F}_{12} \mu_C \mu_E^5 + \mathbf{F}_{13} \mu_C \mu_E^6 \right\|_F^2 \end{aligned} \quad (\text{C.1})$$

where:

$$\begin{aligned} \mathbf{F}_0 &= \mathbf{X}^{(3)} - \mathbf{C}\mathbf{E}_0 & \mathbf{F}_1 &= -\mathbf{C}\mathbf{E}_1 & \mathbf{F}_2 &= -\mathbf{C}\mathbf{E}_2 \\ \mathbf{F}_3 &= -\mathbf{C}\mathbf{E}_3 & \mathbf{F}_4 &= -\mathbf{C}\mathbf{E}_4 & \mathbf{F}_5 &= -\mathbf{C}\mathbf{E}_5 \\ \mathbf{F}_6 &= -\mathbf{C}\mathbf{E}_6 & \mathbf{F}_7 &= -\mathbf{G}_C \mathbf{E}_0 & \mathbf{F}_8 &= -\mathbf{G}_C \mathbf{E}_1 \\ \mathbf{F}_9 &= -\mathbf{G}_C \mathbf{E}_2 & \mathbf{F}_{10} &= -\mathbf{G}_C \mathbf{E}_3 & \mathbf{F}_{11} &= -\mathbf{G}_C \mathbf{E}_4 \\ \mathbf{F}_{12} &= -\mathbf{G}_C \mathbf{E}_5 & \mathbf{F}_{13} &= -\mathbf{G}_C \mathbf{E}_6 \end{aligned}$$

with:

$$\begin{aligned}
\mathbf{E}_0 &= (\mathbf{K}_0^{\odot 2})^\top & \mathbf{E}_1 &= (\mathbf{K}_0 \odot \mathbf{K}_1 + \mathbf{K}_1 \odot \mathbf{K}_0)^\top \\
\mathbf{E}_2 &= (\mathbf{K}_0 \odot \mathbf{K}_2 + \mathbf{K}_1^{\odot 2} + \mathbf{K}_2 \odot \mathbf{K}_0)^\top \\
\mathbf{E}_3 &= (\mathbf{K}_1 \odot \mathbf{K}_2 + \mathbf{K}_2 \odot \mathbf{K}_1 + \mathbf{K}_0 \odot \mathbf{K}_3 + \mathbf{K}_3 \odot \mathbf{K}_0)^\top \\
\mathbf{E}_4 &= (\mathbf{K}_2^{\odot 2} + \mathbf{K}_3 \odot \mathbf{K}_1 + \mathbf{K}_1 \odot \mathbf{K}_3)^\top \\
\mathbf{E}_5 &= (\mathbf{K}_2 \odot \mathbf{K}_3 + \mathbf{K}_3 \odot \mathbf{K}_2)^\top & \mathbf{E}_6 &= (\mathbf{K}_3^{\odot 2})^\top
\end{aligned}$$

and finally:

$$\begin{aligned}
\mathbf{K}_0 &= \mathbf{A}^{(2)} = (\exp(\mathbf{E})) & \mathbf{K}_1 &= \exp(\mathbf{E}) \boxminus \mathbf{G}_\mathbf{E} \\
\mathbf{K}_2 &= (\exp(\mathbf{E}) \boxminus \mathbf{G}_\mathbf{E}^{\boxminus 2})/2 & \mathbf{K}_3 &= (\exp(\mathbf{E}) \boxminus \mathbf{G}_\mathbf{E}^{\boxminus 3})/6
\end{aligned}$$

We define  $\mathbf{F}$  of size  $(I^2 K \times 14)$  as the horizontally concatenation of  $\text{vec}(\mathbf{F}_i)$  for  $0 \leq i \leq 13$ .

#### Appendix D. Simplification of the matrix $\mathbf{Q}$

$\mathbf{Q}$  can be computed from  $\mathbf{F}$  using the expression  $\mathbf{Q} = \mathbf{F}^\top \mathbf{F}$ . However, in terms of numerical complexity, it is better to build  $\mathbf{Q}$  differently. Indeed,  $\mathbf{Q}$  is a symmetric matrix only of size  $(14 \times 14)$ , contrary to  $\mathbf{F}$ , which requires a large storage. Second, the expression of each component of  $\mathbf{Q}$ , except those belonging to the last row and column depending on  $\mathbf{X}_3$ , can be simplified using the following properties

- $(\mathbf{F} \odot \mathbf{G})^\top (\mathbf{F} \odot \mathbf{G}) = \mathbf{F}^\top \mathbf{F} \boxminus \mathbf{G}^\top \mathbf{G}$
- $\text{vec}(\mathbf{FZ}) = (\mathbf{Z}^\top \odot \mathbf{F}) \mathbf{1}_N$

with  $\mathbf{F} \in \mathbb{R}^{P \times N}$ ,  $\mathbf{Z} \in \mathbb{R}^{N \times P}$  and  $\mathbf{G} \in \mathbb{R}^{K \times N}$  [30]. Initially, we have  $Q_{i,j} = \text{vec}(\mathbf{F}_{13-i})^\top \text{vec}(\mathbf{F}_{13-j})$ , for  $0 \leq i, j \leq 12$ . First, using the previous property, both vectors are replaced by a Khatri-Rao product of matrices. Both Khatri-



Rao products are replaced by a Hadamard product. For instance, we get:

$$\begin{aligned}
Q_{1,5} = Q_{5,1} &= \text{Tr}(\mathbf{C}(\mathbf{K}_2^\top \mathbf{K}_0 \boxminus \mathbf{K}_0^\top \mathbf{K}_2) \mathbf{G}_c^\top) & (D.1) \\
&= \text{vec}(\mathbf{G}_c(\mathbf{K}_0 \odot \mathbf{K}_0)^\top)^\top \text{vec}(\mathbf{C}(\mathbf{K}_2 \odot \mathbf{K}_2)^\top) \\
&= \mathbf{1}_p^\top ((\mathbf{K}_0 \odot \mathbf{K}_0) \odot \mathbf{G}_c)^\top ((\mathbf{K}_2 \odot \mathbf{K}_2) \odot \mathbf{C}) \mathbf{1}_p \\
&= \mathbf{1}_p^\top ((\mathbf{K}_0 \odot \mathbf{K}_0)^\top (\mathbf{K}_2 \odot \mathbf{K}_2) \boxminus \mathbf{G}_c^\top \mathbf{C}) \mathbf{1}_p \\
&= \mathbf{1}_p^\top (\mathbf{K}_2^\top \mathbf{K}_0 \boxminus \mathbf{K}_2^\top \mathbf{K}_0 \boxminus \mathbf{G}_c^\top \mathbf{C}) \mathbf{1}_p & (D.2)
\end{aligned}$$

Note that this way of simplifying the matrix trace was originally proposed by Tomasi and Bro [3]. The same technique is used for the other components except those depending on  $\mathbf{X}_3$  and by which the numerical complexity is governed.

- [1] R. A. HARSHMAN, Foundation of PARAFAC procedure: Models and conditions for an 'explanatory' multi-mode factor analysis, UCLA working papers in Phonetics (16) (1970) 1–84.
- [2] H. BECKER, P. COMON, L. ALBERA, M. HAARDT, I. MERLET, Multi-way space-time-wave-vector analysis for EEG source separation, Signal Processing.
- [3] G. TOMASI, R. BRO, A comparison of algorithms for fitting the Parafac model, Computational Statistics and Data Analysis 50 (7) (2006) 1700–1734.
- [4] J. CARROLL, J. CHANG, Analysis of individual differences in multidimensional scaling via an n-way generalization of eckart-young decomposition, Psychometrika 35 (9) (1970) 267–283.
- [5] Y. TAKANE, Applications of multidimensional scaling in psychometrics., In Rao, C. R., and Sinharay, S. (Eds.), Handbook of Statistics (Vol. 26): Psychometrics, (pp. 359-400). Amsterdam: Elsevier., 2007.
- [6] F. HUSSON, J. PAGES, INDSCAL model: geometrical interpretation and methodology, Computational Statistics and Data Analysis 50 (2) (2006) 358 – 378. doi:10.1016/j.csda.2004.08.005.

- [7] A. YEREDOR, Non-orthogonal joint diagonalization in the least-squares sense with application in blind source separation, *IEEE Transactions on Signal Processing* 50 (7) (2002) 1545–1553.
- [8] A. BELOUCHRANI, K. ABED-MERAIM, J.-F. CARDOSO, E. MOULINES, A blind source separation technique using second-order statistics, *IEEE Transactions On Signal Processing* 45 (2) (1997) 434–444.
- [9] J.-F. CARDOSO, A. SOULOUMIAC, Blind beamforming for non-gaussian signals, *IEE Proceesings-F* 140, no. 6 (1993) 362–370.
- [10] L. ALBERA, P. COMON, L. C. PARRA, A. KARFOUL, A. KACHENOURA, L. SENHADJI, Comparative performance study of independent component analysis methods on biomedical signals, in: *Handbook of Blind Source Separation*, P. Comon and C. Jutten Eds, Academic Press, 2010., 2010.
- [11] A. KACHENOURA, L. ALBERA, L. SENHADJI, P. COMON, ICA: a potential tool for BCI systems, *IEEE Signal Processing Magazine*, special issue on Brain-Computer Interfaces 25 (1) (2008) 57–68.
- [12] F. ASANO, S. IKEDA, M. OGAWA, H. ASOH, N. KITAWAKI, Combined approach of array processing and independent component analysis for blind separation of acoustic signals, *IEEE Transactions On Speech and Audio Processing* 11 (3) (2003) 204–215.
- [13] C. LADROUE, F. A. HOWE, J. R. GRIFFITHS, A. R. TATE, Independent component analysis for automated decomposition of in vivo magnetic resonance spectra, *Magnetic Resonance in Medecine* 50 (2003) 697–703.
- [14] D.-T. PHAM, Joint approximate diagonalization of positive definite matrices, *SIAM Journal on Matrix Analysis Appl.* 22 (4) (2001) 1136–1152.
- [15] B. AFSARI, Sensitivity analysis for the problem of matrix joint diagonalization, *SIAM J. Matrix Analysis and Applications* 30 (3) (2008) 1148–1171.

- [16] T. G. KOLDA, B. W. BADER, Tensor decompositions and applications, *SIAM Review* 51 (3) (2009) 455–500.
- [17] J. T. BERGE, H. KIERS, Some clarifications of the candecomp algorithm applied to indscal, *Psychometrika* 56 (1991) 317–326.
- [18] D.-G. KANG, Y. SUNG, S. KIM, S. LEE, C. KIM, Multiple object decomposition based on independent component analysis of multi-energy x-ray projections., in: 16th IEEE International Conferences on Image Processing (ICIP), 2009, pp. 4173–4176.
- [19] H. PARASTAR, M. JALALI-HERAVI, R. TAULER, Is independent component analysis appropriate for multivariate resolution in analytical chemistry ?, *J. Analytical Chemistry* 31 (2012) 134–143.
- [20] J. HAO, X. ZOU, P. M. WILSON, N. P. DAVIES, Y. SUN, C. A. PEET, T. N. ARVANITIS, A comparative study of feature extraction and blind source separation of independent component analysis (ICA) on childhood brain tumour 1H magnetic resonance spectra, *NMR in Biomedicine* 22 (8) (2009) 809–818.
- [21] J. PULKKINEN, A.-M. HAKKINEN, N. LUNDBOM, A. PAETAU, R. A. KAUPPINEN, Y. HIKTUNEN, Independent component analysis to proton spectroscopic imaging data of human brain tumours, *European Journal of Radiology* 56 (2005) 160–164.
- [22] M. S. BARTLETT, J. MOVELLAN, T. SEJNOWSKI, Face recognition by Independent Component Analysis, *IEEE Transactions on Neural Networks* 13 (6) (2002) 1450–1464.
- [23] B. CHEN, O. ACOSTA, A. KACHENOURA, J. D. OSPINA, G. DREAN, A. SIMON, J. J. BELLANGER, P. HAIGRON, R. D. CREVOISIER, Spatial characterization and classification of rectal bleeding in prostate cancer radiotherapy with a voxel-based principal components analysis model for 3D dose distribution, in: in Prostate Cancer Imaging Workshop, MICCAI

2011, Toronto, CA, Sep 2011, Lecture Notes in Computer Science, Vol. 6963, 2011, pp. 60–69.

- [24] J. COLOIGNER, L. ALBERA, A. KACHENOURA, L. SENHADJI, Canonical décomposition of semi-symmetric semi-nonnegative three-way arrays, in: ASILOMAR, 2011 IEEE Forty-Fifth Asilomar Conference on Signals, Systems and Computers, Asilomar, US, 2011.
- [25] J. COLOIGNER, L. ALBERA, A. KARFOUL, A. KACHENOURA, P. COMON, L. SENHADJI., Semi-nonnegative independent component analysis, in: in LVA/ICA 10, ninth international conference on latent variable analysis and signal separation, Saint Malo, France, 2010, pp. 612–619.
- [26] J.-P. ROYER, N. THIRION-MOREAU, P. COMON, Computing the polyadic decomposition of nonnegative third order tensors, *Signal Processing* 91 (9) (2011) 2159–2171. doi:10.1016/j.sigpro.2011.03.006.
- [27] M. RAJIH, P. COMON, R. A. HARSHMAN, Enhanced line search : A novel method to accelerate PARAFAC, *SIAM Journal on Matrix Analysis Appl.* 30 (3) (2008) 1148–1171. doi:10.1137/06065577.  
URL <http://link.aip.org/link/?SML/30/1128/1>
- [28] P. COMON, X. LUCIANI, A. L. F. D. ALMEIDA, Tensor decomposition, alternating least squares and other tales, *Journal of Chemometrics* 23 (393–405) (2009) 393–405.
- [29] R. BRO, S. D. JONG, A fast non-negativity-constrained least squares algorithm, *Journal of Chemometrics* 11 (1999) 393–401.
- [30] J. W. BREWER, Kronecker products and matrix calculus in system theory, *IEEE Transactions On Circuits and Systems* 25 (9) (1978) 114–122.
- [31] J. NOCEDAL, S. J. WRIGHT, Numerical optimization second edition, Springer, 2006.

- [32] Y. CHEN, D. HAN, L. QI, New als methods with extrapolation search direction and optimal step size for complex-valued tensor decomposition, *IEEE Transactions on Signal Processing* 59 (2011) 5888–5898.
- [33] A. FRANC, Etude algebrique des multitableaux: Apport de l’algebre tensorielle, Ph.D. thesis, Universit  de Montpellier 2 (1992).
- [34] J. COLOIGNER, A. KARFOUL, L. ALBERA, P. COMON, Line search and trust region strategies for canonical decomposition of semi-nonnegative semi-symmetric 3rd order tensors, *Linear Algebra and its applications* 450 (2014) 334–374. doi:<http://dx.doi.com/10.1016/j.laa.2014.02.001>.
- [35] S. MOUSSAOUI, D. BRIE, A. MOHAMAD-DJAFARI, C. CARTERET, Separation of non-negative mixture of non-negative sources using a bayesian approach and mcmc sampling, *IEEE Transactions on Signal Processing* 54 (2006) 4133–4145.
- [36] A. HYVARINEN, Fast and robust fixed-point algorithms for independent component analysis, *IEEE Transactions On Neural Networks* 10, no. 3 (1999) 626–634.
- [37] E. MOREAU, A generalization of joint-diagonalization criteria for source separation, *IEEE on Signal Processing* 49 (3) (2001) 530–541.
- [38] P. COMON, Independent Component Analysis, a new concept ?, *Signal Processing, Elsevier* 36 (3) (1994) 287–314.
- [39] L. ALBERA, A. FERREOL, P. COMON, P. CHEVALIER, Blind Identification of Overcomplete Mixtures of sources (BIOME), *Linear Algebra Applications* 391C (2004) 3–30.
- [40] F. SAMARIA, A. HARTER, Parameterisation of a stochastic model for human face identification., in: 2nd IEEE Workshop on Applications of Computer Vision, Sarasota (Florida), 1994.

- [41] A. HJORUNGNES, D. GESBERT, Complex-valued matrix differentiation: Techniques and key results, IEEE Transaction on Signal Processing 55 (6) (2007) 2740–2746.
- [42] W. BRANSON, Solving the cubic with cardano, Tech. rep., Mathematical association of america (2013).
- [43] P. PAATERO, A weighted non-negative least squares algorithm for three-way 'parafac' factor analysis, Chemometrics and Intelligent Laboratory Systems 38 (1997) 223–242.